



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/510,167	07/18/2005	Gunnar Nordmark	1533-1005	6777
<div>465 7590 08/16/2010</div> <div>YOUNG &amp; THOMPSON 209 Madison Street Suite 500 Alexandria, VA 22314</div>				
EXAMINER				
SCLACCA, SCOTT M				
ART UNIT		PAPER NUMBER		
2446				
NOTIFICATION DATE		DELIVERY MODE		
08/16/2010		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

DocketingDept@young-thompson.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/510,167  
Filing Date: July 18, 2005  
Appellant(s): NORDMARK ET AL.

James J. Livingston, Jr. (Reg. No. 55,394)  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed on May 27, 2010 appealing from the Office action mailed on August 20, 2009.

**(1) Real Party in Interest**

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The following is a list of claims that are rejected and pending in the application:

Claims 1, 5-9 and 11-22 are pending and stand rejected.

**(4) Status of Amendments After Final**

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

**(5) Summary of Claimed Subject Matter**

The examiner has no comment on the summary of claimed subject matter contained in the brief.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

**(7) Claims Appendix**

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

**(8) Evidence Relied Upon**

US 2002/0122424	Kawarai et al.	9-2002
US 2003/0046429	Sonksen	3-2003
US 5,818,894	Song	10-1998
US 6,996,117	Lee et al.	2-2006
US 7,016,531	Farinacci et al.	3-2006
WO 99/60708	Hultsch	11-1999

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1, 5-9, 11-12, 15-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawarai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708).

Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawarai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708) and further in view of Lee et al. (US 6,996,117).

Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawarai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708) and further in view of Song (US 5,818,894).

***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1, 5-9, 11-12, 15-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708).

Regarding Claim 1, Sonksen teaches a method of pipelined processing of a data packet in a processing means comprising at least two processing stages (*"A method and apparatus for packet processing is disclosed. In one embodiment of the invention the method and apparatus is implemented in plurality of pipeline stages"* – See Abstract), said data packet containing information (*"In most instances, each packet includes a payload portion, containing the data, preceded by a header portion, containing information about the packet such as the source and destination of the packet, quality of service, packet size and other information"* – See [0002]; The packet includes a payload (information) portion), said method comprising:

generating an intermediate data packet by adding at least one of a dummy header and dummy tail to said data packet, said dummy header and dummy tail being capable of storing information of a communication system (*"the invention may comprise a method of adding a tag to a packet comprising identifying a control word to guide processing of a packet and then storing a portion of a packet in a memory"* – See [0015]);

associating information reference to said intermediate data packet (*"each packet is associated with a control word"* – See [0025]; *"The control words are generated from information extracted from the packet handle"* – See [0095]; *"The packet handle*

*comprises information about the packet, such as including but not limited to the packet format, packet length or type of service” – See [0094]);*

*storing said information reference in additional register (“In one embodiment the invention comprises a system for dynamically modifying or supplementing the contents of a packet on a packet by packet basis including a first memory configured to store a control word” – See [0017]); and*

*processing said intermediate data packet in a processing stage (“the invention modifies a packet based on control instructions and includes a pipeline processing stage comprising one or more memory modules configured to store packet data and supplemental data” – See [0019]).*

Sonksen goes on to disclose that various fields in a packet may be modified (*“The one or more data modifiers may perform modifications consisting of modification to a time to live value, a type of service value, a checksum value, or other data fields in a packet” – See [0022]*). Sonksen shows that values such as a time to live value or checksum value must be updated as a packet is processed in order to account for the modifications to the information in the packet (*“In this embodiment the checksum generator 1340 creates a running total of the value of the bytes in a packet. The use of a checksum for error checking is know in the art and accordingly not described in great detail here. If a packet is modified it may be desired to modify the checksum to account for the modification” – See [0112]*). Thus, Sonksen generally teaches as shown above that certain fields in a packet are modified to account for changes to the information in the packet. Sonksen does not explicitly teach that the information reference comprises

information relating to the length and position of the information of the data packet contained in the intermediate data packet. However, Kawai discloses a packet containing parameters regarding a length and position of the information (payload portion) of a data packet (*"Offset Value indicates the start position of a valid field for payload information in a fixed-length packet. Payload Length indicates the length of the valid field from that start position"* – See [0053]). It would have been obvious to one of ordinary skill in the art at the time the invention was made to associate with a packet the length and position of information within said packet. Motivation for doing so would be to provide information regarding the location of the "useful" data otherwise known as the payload of a packet so that the packet may be handled properly when received (See Kawai, [0056]).

Sonksen does not explicitly teach the change in length of information of the data packet comprising:

adding to the length a value representing a length of a portion of the at least one of the dummy header and dummy tail of said intermediate data packet, the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet representing empty information after said processing.

However, Farinacci does teach a change in the length of information of a data packet comprising adding to the length a value representing a length of a portion of the at least one of the dummy header and dummy tail of said intermediate data packet, the portion storing information of the communications system (See Fig. 2B; *"Length 204, or*

*No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list (a) is therefore ceiling((6+n)/4) 32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is (a+n)”— See Col. 5, lines 9-14; “SGM capable routers append their IP address to the list, update the offset and re-forward the packet”— See Col. 15, lines 48-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the length a value representing a length of a portion of the dummy header of a data packet. Motivation for doing so would be to make forwarding decisions more efficient (See Farinacci, Col. 5, lines 24-25).*

Kawarai teaches subtracting from the length a value representing a portion of said intermediate data packet representing empty information after processing (*“Just before the variable-length packet being output from an output line, the overhead portions of the plurality of fixed-length packets are removed to change the variable-length packet into the original complete one”— See [0119]*).

Sonksen further teaches upon said intermediate data packet exiting the last of said at least one processing stages removing data from said intermediate data packet (*“The dynamic stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a more flexible control structure. The static stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a hardwired system”— See Abstract; As shown in Fig. 5A the*

pipeline includes at least two stages, one dynamic and one static, both of which may remove bits from a packet).

Sonksen does not explicitly teach the removal being based on a determination of whether any bits of said intermediate data packet are superfluous. However, Hultsch discloses removing superfluous data from a packet (*"removing the superfluous filler data received via the circuit-switched connection in the data stream (DS1) with the constant data rate and by reformatting the useful data for the data stream with the variable data rate and sending it via a packet-orientated connection"* – See Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include removing superfluous bits from said intermediate data packet. Motivation for doing so would be to make efficient use of transmission bandwidth (See Hultsch, Abstract).

Regarding Claim 5, Sonksen teaches removing, upon said intermediate data packet exiting the last of said at least one processing stages, at least one bit from said intermediate data packet (*"The dynamic stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a more flexible control structure. The static stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a hardwired system"* – See Abstract; As shown in Fig. 5A the pipeline includes at least two stages, one dynamic and one static, both of which may remove bits from a packet).

Regarding Claim 6, Sonksen teaches said information reference being included in additional information associated with said intermediate data packet (*"Note that in different formats or protocols the entire header may be of different length, items of information may be arranged in different order, pieces of information in the header may be of different length, or additional items of information may be included. Advantageously the dynamic processing module of the invention is able to dynamically adapt and/or accommodate the various different header formats and the challenges presented by different formats"* – See [0081]; As shown above with regard to Claim 1, the information reference (position and length of an information item within a packet) is taken into consideration when the corresponding packet is processed).

Regarding Claim 7, Sonksen further teaches that prior to said step of processing said intermediate data packet, said information reference is stored in at least one register accessible to the processing stage performing said processing (*"The method may also include storing associated packets and control words prior to entry of a packet into the processing pipeline"* – See [0033]).

Regarding Claim 8, the combination of Sonksen and Kawai teaches the information reference comprising a length value and an offset value, said length value representing the length of the information contained in said intermediate data packet

and said offset value indicating the position in said intermediate data packet of the information contained in said data packet as shown above with respect to Claim 1.

Regarding Claim 9, Sonksen teaches a processing means for pipelined processing of a data packet (*"FIG. 3 illustrates a block diagram of an example embodiment of a pipeline packet processing system of the invention"* – See [0037]), said processing means comprising at least one processing stage comprising a logic unit (*"Shown in FIG. 3 is a first processing module 312, a second processing module 320 up to an Nth processing module 328"* – See [0062]) and a register for storing at least part of said data packet (*"In one embodiment the input line 300 may carry packets and associated control data to the memory 308 for storage"* – See [0061]), said processing means comprising:

a receiver is adapted to receive said data packet and to generate an intermediate data packet by adding at least one of a dummy header and a dummy tail to said data packet, said dummy header and dummy tail being capable of storing information of a communications system (*"The dynamic processing module 502 comprises a module that may be configured to generate or modify a tag or other portion of a packet header, such as a tag that may be attached to a portion of a packet to aid in packet processing or routing"* – See [0068]);

at least one register for storing information reference associated with said intermediate data packet is accessible to said logic unit (*"In one embodiment the invention comprises a system for dynamically modifying or supplementing the contents*

*of a packet on a packet by packet basis including a first memory configured to store a control word*” – See [0017]; *“According, a first controller 316 connects or communicates with the memory 308 and the first processing module 312”* – See [0062]); and

at least one of at said at least one logic units is adapted to operate upon said information reference (*“The first processing module 312 processes the data in accordance with the controller 316”* – See [0063]).

Sonksen does not explicitly teach the information reference comprising information relating to the length and position of the information of the data packet contained in said intermediate data packet. However, Kawarai discloses a packet containing parameters regarding a length and position of the information (payload portion) of a data packet (*“Offset Value indicates the start position of a valid field for payload information in a fixed-length packet. Payload Length indicates the length of the valid field from that start position”* – See [0053]).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to associate with a packet the length and position of information within said packet for the same reasons as those given with respect to Claim 1.

Sonksen teaches a processing stage for processing said intermediate data packet (*“As data is finished being processed by the first processing module 312, the data progresses down the path to the second processing module 320”* – See [0063]).

Sonksen does not explicitly teach the length of the information of the data packet contained in said intermediate data packet changing upon processing said intermediate

data packet in said processing stage, whereby said information reference is altered in order for said information reference to reflect said change,

wherein the change in the length of said information of said data packet comprises at least one of:

adding to the length a value representing a length of the portion of the at least one of the dummy header and dummy tail of said intermediate data packet, the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet comprising empty information after said processing.

However, Farinacci does teach the length of the information contained in a data packet changing upon processing said data packet, whereby said information reference is altered in order for said information reference to reflect said change,

wherein the change in the length of said information of said data packet comprises:

adding to the length a value representing a length of the portion of a dummy header of said data packet, the portion storing information of the communications system (See Fig. 2B; “Length 204, or No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list ( $a$ ) is therefore  $\text{ceiling}((6+n)/4)$  32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is  $(a+n)$ ” — See Col. 5, lines 9-14; “SGM capable routers append their IP address to the list, update the offset and re-forward the packet” — See Col. 15, lines 48-50). It would have been obvious to one of

ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the length a value representing a length of a portion of the dummy header of a data packet for the same reasons as those given with respect to Claim 1.

Kawarai teaches subtracting from the length a value representing a portion of said intermediate data packet representing empty information after processing (*"Just before the variable-length packet being output from an output line, the overhead portions of the plurality of fixed-length packets are removed to change the variable-length packet into the original complete one"* – See [0119]).

Sonksen further teaches upon said intermediate data packet exiting the last of said at least one processing stages removing data from said intermediate data packet (*"The dynamic stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a more flexible control structure. The static stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a hardwired system"* – See Abstract; As shown in Fig. 5A the pipeline includes at least two stages, one dynamic and one static, both of which may remove bits from a packet).

Sonksen does not explicitly teach the removal being based on a determination of whether any bits of said intermediate data packet are superfluous. However, Hultsch discloses removing superfluous data from a packet (*"removing the superfluous filler data received via the circuit-switched connection in the data stream (DS1) with the constant data rate and by reformatting the useful data for the data stream with the*

*variable data rate and sending it via a packet-orientated connection*" – See Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include removing superfluous bits from said intermediate data packet for the same reasons as those given with respect to Claim 1.

Regarding Claim 11, Sonksen teaches said receiver for adding comprising a buffer (*"FIG. 8 illustrates an example implementation of one example embodiment of the tag generation module"* – See [0043]; *"In the embodiment shown in FIG. 8, the FIFO unit 800 includes control word/label storage 804 and packet data storage 806"* – See [0083]) and a shifter (*"In the example embodiment shown in FIG. 8, the second register 844 includes four byte shift registers"* – [0087]).

Regarding Claim 12, Sonksen teaches the means of Claim 9 further comprising means for removing at least one bit from said intermediate data packet (*"It also includes a processing module configured to add supplemental data to a packet or strip data from a packet based on control instructions and the processing location in the packet"* – See [0019]).

Regarding Claim 15, Sonksen teaches said at least one register for storing information reference being located in said processing stage (Each processing module in Fig. 3 includes an interface to memory 308).

Regarding Claim 16, Sonksen teaches said at least one register for storing information reference comprising one register and another register (*"This system may include a control word bank having two or more locations"* – See [0025]). As shown above with regard to Claim 1, Hultsch teaches that the values may be a length value and an offset value.

Regarding Claim 17, Sonksen teaches an integrated circuit characterized by processing means according to Claim 9 (*"In one embodiment an the method and apparatus is enabled in an ASIC-based solution"* – See [0012]; The processing means is characterized by an ASIC (application-specific integrated circuit)).

Regarding Claim 18, Sonksen teaches a computer unit characterized by an integrated circuit according to Claim 9 (*"In one embodiment an the method and apparatus is enabled in an ASIC-based solution"* – See [0012]; The processing means is characterized by an ASIC (application-specific integrated circuit)).

Regarding Claim 19, Sonksen teaches a pipelined processor for processing a data packet, comprising:

a register for storing at least part of the data packet (*"invention may comprise a method of adding a tag to a packet comprising identifying a control word to guide processing of a packet and then storing a portion of a packet in a memory"* – See [0015]);

at least one additional register for storing an information reference for association with an intermediate data packet (*"In one embodiment the invention comprises a system for dynamically modifying or supplementing the contents of a packet on a packet by packet basis including a first memory configured to store a control word"* – See [0017]);

a logic unit performing the steps of:

receiving the data packet (*"an input line 300 receives packets and/or control information to be processed by the pipeline operation"* – See [0061]);

generating the intermediate data packet by adding at least one of a dummy header and dummy tail to the data packet (*"The dynamic processing module 502 comprises a module that may be configured to generate or modify a tag or other portion of a packet header, such as a tag that may be attached to a portion of a packet to aid in packet processing or routing"* – See [0068]);

associating information reference to the intermediate data packet (*"each packet is associated with a control word"* – See [0025]; *"The control words are generated from information extracted from the packet handle"* – See [0095]; *"The packet handle comprises information about the packet, such as including but not limited to the packet format, packet length or type of service"* – See [0094]);

storing the information reference in the at least one additional register (*"In one embodiment the invention comprises a system for dynamically modifying or supplementing the contents of a packet on a packet by packet basis including a first memory configured to store a control word"* – See [0017]); and

processing the intermediate data packet in a processing stage (*"The first processing module 312 processes the data in accordance with the controller 316"* – See [0063]).

Sonksen goes on to disclose that various fields in a packet may be modified (*"The one or more data modifiers may perform modifications consisting of modification to a time to live value, a type of service value, a checksum value, or other data fields in a packet"* – See [0022]). Sonksen shows that values such as a time to live value or checksum value must be updated as a packet is processed in order to account for the modifications to the information in the packet (*"In this embodiment the checksum generator 1340 creates a running total of the value of the bytes in a packet. The use of a checksum for error checking is know in the art and accordingly not described in great detail here. If a packet is modified it may be desired to modify the checksum to account for the modification"* – See [0112]). Thus, Sonksen generally teaches as shown above that certain fields in a packet are modified to account for changes to the information in the packet. Sonksen does not explicitly teach that the information reference comprises information relating to the length and position of the information of the data packet contained in the intermediate data packet. However, Kawai discloses a packet containing parameters regarding a length and position of the information (payload portion) of a data packet (*"Offset Value indicates the start position of a valid field for payload information in a fixed-length packet. Payload Length indicates the length of the valid field from that start position"* – See [0053]). It would have been obvious to one of ordinary skill in the art at the time the invention was made to associate with a packet the

length and position of information within said packet for the same reasons as those given with respect to Claim 1.

Sonksen does not explicitly teach the change in length of information of the data packet comprising:

adding to the length a value representing a length of a portion of the at least one of the dummy header and dummy tail of said intermediate data packet, the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet comprising empty information after said processing.

However, Farinacci does teach a change in the length of information of a data packet comprising adding to the length a value representing a length of a portion of the at least one of the dummy header and dummy tail of said intermediate data packet, the portion storing information of the communications system (See Fig. 2B; “Length 204, or No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list (a) is therefore ceiling((6+n)/4) 32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is (a+n)” — See Col. 5, lines 9-14; “SGM capable routers append their IP address to the list, update the offset and re-forward the packet” — See Col. 15, lines 48-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the length a value representing a length of a portion of the dummy header of a data packet for the same reasons as those given with respect to Claim 1.

Kawarai teaches subtracting from the length a value representing a portion of said intermediate data packet representing empty information after processing (*“Just before the variable-length packet being output from an output line, the overhead portions of the plurality of fixed-length packets are removed to change the variable-length packet into the original complete one”* – See [0119]).

Sonksen further teaches upon said intermediate data packet exiting the last of said at least one processing stages removing data from said intermediate data packet (*“The dynamic stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a more flexible control structure. The static stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a hardwired system”* – See Abstract; As shown in Fig. 5A the pipeline includes at least two stages, one dynamic and one static, both of which may remove bits from a packet).

Sonksen does not explicitly teach the removal being based on a determination of whether any bits of said intermediate data packet are superfluous. However, Hultsch discloses removing superfluous data from a packet (*“removing the superfluous filler data received via the circuit-switched connection in the data stream (DS1) with the constant data rate and by reformatting the useful data for the data stream with the variable data rate and sending it via a packet-orientated connection”* – See Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was

made to modify Sonksen to include removing superfluous bits from said intermediate data packet for the same reasons as those given with respect to Claim 1.

Regarding Claim 20, Sonksen in view of Kawarai and Farinacci teaches the method of Claim 1. Sonksen teaches processing of said intermediate data packet (*"As data is finished being processed by the first processing module 312, the data progresses down the path to the second processing module 320. Thus, the second processing module 320 conducts processing on data simultaneous with other data processing being processed by the first processing module 312"* – See [0063]).

Sonksen does not explicitly teach the processing resulting in a change of the position of said information of said data packet contained in said intermediate data packet, whereby said information reference is altered in order for said information reference to reflect said change, and wherein the change in the position of said information of said data packet comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header of said intermediate data packet containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

However, Farinacci does teach processing resulting in a change of the position of information contained in a data packet, whereby an information reference is altered in order for said information reference to reflect said change, and wherein the change in

the position of said information of said data packet comprises adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing (See Fig. 2B; *"Length 204, or No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list (a) is therefore ceiling((6+n)/4) 32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is (a+n)"* – See Col. 5, lines 9-14; *"SGM capable routers append their IP address to the list, update the offset and re-forward the packet"* – See Col. 15, lines 48-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the position a value representing a portion of an intermediate data packet representing empty information for the same reasons as those given with respect to Claim 1.

Regarding Claim 21, Sonksen in view of Kawarai and Farinacci teaches the processing means of Claim 9. Sonksen teaches a processing stage for processing said intermediate data packet (*"As data is finished being processed by the first processing module 312, the data progresses down the path to the second processing module 320. Thus, the second processing module 320 conducts processing on data simultaneous with other data processing being processed by the first processing module 312"* – See [0063]).

Sonksen does not explicitly teach that the processing stage is configured to process said intermediate data packet resulting in a change of the position of said

information of said data packet contained in said intermediate data packet, whereby said information reference is altered in order for said information reference to reflect said change, and wherein the change in the position of said information of said data packet comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header of said intermediate data packet containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

However, Farinacci does teach a processing stage configured to process a data packet resulting in a change of the position of information contained in the data packet, whereby an information reference is altered in order for said information reference to reflect said change, and wherein the change in the position of said information of said data packet comprises adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing (See Fig. 2B; “Length 204, or No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list ( $a$ ) is therefore  $\text{ceiling}((6+n)/4)$  32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is  $(a+n)$ ” — See Col. 5, lines 9-14; “SGM capable routers append their IP address to the list, update the offset and re-forward the packet” — See Col. 15, lines 48-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the position a value

representing a portion of an intermediate data packet representing empty information for the same reasons as those given with respect to Claim 1.

Regarding Claim 22, Sonksen in view of Kawarai and Farinacci teaches the pipelined processor of Claim 19. Sonksen teaches a processing stage for processing said intermediate data packet (*"As data is finished being processed by the first processing module 312, the data progresses down the path to the second processing module 320. Thus, the second processing module 320 conducts processing on data simultaneous with other data processing being processed by the first processing module 312"* – See [0063]).

Sonksen does not explicitly teach that the processing stage is configured to process said intermediate data packet resulting in a change of the position of said information of said data packet contained in said intermediate data packet, whereby said information reference is altered in order for said information reference to reflect said change, and wherein the change in the position of said information of said data packet comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header of said intermediate data packet containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

However, Farinacci does teach a processing stage configured to process a data packet resulting in a change of the position of information contained in the data packet, whereby an information reference is altered in order for said information reference to reflect said change, and wherein the change in the position of said information of said data packet comprises adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing (See Fig. 2B; “Length 204, or No. of nodes.—The number of addresses in the address list (1 byte). The offset to the start of the address list (a) is therefore  $\text{ceiling}((6+n)/4)$  32 bit words and hence the total length of the SGM header (i.e. the offset to the start of the encapsulated multicast data packet) is  $(a+n)$ ” – See Col. 5, lines 9-14; “SGM capable routers append their IP address to the list, update the offset and re-forward the packet” – See Col. 15, lines 48-50). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include adding to the position a value representing a portion of an intermediate data packet representing empty information for the same reasons as those given with respect to Claim 1.

3. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawarai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708) and further in view of Lee et al. (US 6,996,117).

Regarding Claim 13, Sonksen teaches means for removing comprising a buffer (“FIG. 17 illustrates an example embodiment of a data modifier system as may be

*contemplated for use with a pipeline processing system*” – See [0054]; “Register 1704 has sections A, B, C and D” – See [0125]), but does not explicitly teach means for removing also comprising a shifter. However, Lee does teach using a shifter as means for removing data (“*The stripped-off information element segment 450 is rotated backward by one place (i.e., shifted to the left by one place) to produce the rotated information element segment 452*” – See Col. 29, lines 65-67). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a shifter as means for removing at least one bit of data. Motivation for doing so would be to implement as much of the processing means as possible in hardware in order to boost performance (See Col. 1, lines 65-67 of Lee's disclosure).

4. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sonksen (US 2003/0046429) in view of Kawarai et al. (US 2002/0122424), Farinacci et al. (US 7,016,351) and Hultsch (WO 99/60708) and further in view of Song (US 5,818,894).

Regarding Claim 14, Sonksen teaches means for adding comprising a shifter (See above remarks regarding Claim 11), but does not explicitly teach the shifter being a barrel shifter. However Song does teach a barrel shifter with inputs for adding input data (“*A high speed barrel shifter in which fill input data is especially added*” – See Abstract; “*FIG. 1 is a block diagram of an 8-bit barrel shifter according to the present invention*” – See Col. 1, lines 61-62). It would have been obvious to one of ordinary skill in the art at the time the invention was made to include a barrel shifter as means for

adding at least one bit to a packet of data. Motivation for doing so would be to provide a device which can perform shift operations at high speed while minimizing the necessary logic circuitry (See Col. 1, lines 25-27 of Song's disclosure).

#### **(10) Response to Argument**

The examiner summarizes the various points raised by the appellant and addresses them individually.

**(A) Appellant Argues:** "However, the Applicants respectfully submit that the combination of the references do not teach the removal of superfluous bits from headers and/or tails of a packet.

It is noted that the rejection never mentions the removal of superfluous bits from the header or the tail. It would appear that the Office implicitly includes the header and tail as part of the payload of the packet and therefore indistinguishable from the payload. The Applicants respectfully disagree with this interpretation. It is well known in the art that a header defines information about the packet such as its source IP address, destination IP address, Time To Live, checksum information, etc..., (See Fig. 9B of Sonksen) while the payload is the data to be transmitted. In a classic sense routers and other network equipment need not know anything about the payload of a packet, while the header is relied on for the function of network equipment."

***In Response:*** Appellant has provided valid examples of information that may be included in a packet header or tail (i.e., source/destination addresses, TTL values, checksums, etc.). However, this is not a limiting example of the types of information that are included in a header or tail. The information conveyed in a packet header varies depending on the network protocols used or the type of network the packet is being transmitted over. According to the broadest reasonable interpretation, a packet header can be any information or field in a packet other than the payload data field that is used for conveying some type of control information.

Sonksen discloses that portions of a packet may be modified, removed or supplemented (*"The dynamic stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a more flexible control structure. The static stages are configured to modify, remove, or supplement portions of the packet as the packet or portion thereof passes through the pipeline with the aid of a hardwired system"* – See Abstract). In further detail with regard to the feature of removing information from a packet, Sonksen discloses removing a tag (*"One method of increasing routing speed is to attach additional data in front of the header and use this information to efficiently route a packet. This additional data is often referred to as a tag. The tag is formed of protocol data that may be utilized by the router to more rapidly analyze and route the packet"* – See [0005]; *"In the exemplary method of FIG. 11B a memory controller module loads packet data and packet control word(s) into the FIFO unit or buffer. The control word*

*may contain label data. In this example method the dynamic processing module generates, modifies, or removes a tag” – See [0094]).*

Although the tag is made up of supplemental data that is attached to a pre-existing header, the tag is considered to be part of a header because it does not contain any payload data and it includes control information that facilitates analysis and routing of the packet in much the same way that the header fields of an Internet Protocol packet, such as source IP address, destination IP address and Time To Live values facilitate analysis and routing of IP packets.

As admitted in the Final Office Action dated August 20, 2009, Sonksen does not explicitly teach that the determination to remove the tag from a packet header is based on whether the bits of information included in the tag are determined to be superfluous. However, Hultsch discloses removing superfluous data from a packet (*“removing the superfluous filler data received via the circuit-switched connection in the data stream (DS1) with the constant data rate and by reformatting the useful data for the data stream with the variable data rate and sending it via a packet-orientated connection” – See Abstract*). Modifying Sonksen in view of the teachings of Hultsch provides the advantage of efficiently utilizing available network bandwidth by not wasting it transmitting data that is determined to be irrelevant or otherwise of no use.

The same reasoning given above applies to Appellant’s arguments with respect to Claims 9 and 19.

**(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Scott M. Sciacca/

Examiner, Art Unit 2446

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451

Conferees:

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2446

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2451